Reputation for Inter-Domain QoS Routing

Emmanuelle Anceaume, Yann Busnel, Paul Lajoie-Mazenc, Géraldine Texier

Reputation for Inter-Domain QoS Routing

Emmanuelle Anceaume
IRISA / CNRS,
Rennes, France
anceaume@irisa.fr

Yann Busnel
Crest (Ensai) / Inria
Rennes, France
Yann.Busnel@ensai.fr

Paul Lajoie-Mazenc
IRISA / Université de Rennes 1,
Rennes, France
paul.lajoie-mazenc@irisa.fr

Géraldine Texier
IRISA / Telecom Bretagne,
Rennes, France
geraldine.texier@telecom-bretagne.eu

***Abstract*—Video traffic, which represents an increasing fraction of the Internet traffic, requires end-to-end quality of service (QoS) guarantees for inter-domain routing. However, providing such guarantees remains a challenge essentially because it requires a strong and fair cooperation among the different network operators or Autonomous Systems (ASes), crossed by the traffic. Having a single AS on the path that does not meet its QoS engagement is sufficient to violate the end-to-end QoS guarantees. Unfortunately, the client is not capable of distinguishing unfair ASes from honest ones at the time it selects its path. Reputation mechanisms turn out to be very efficient tools to estimate how trustworthy and reliable entities can be without requiring the help of any central authority. They are effective to foster cooperation by remedying selfishness. In this position paper, we identify the main properties a reputation mechanism should meet to improve inter-domain QoS routing, and we provide a coarsed-grain vision of the design of such a mechanism.**

Keywords*—*Inter-Domain Path Computation; Quality of Service; Routing; Reputation Mechanism; Distributed Systems

I. Introduction

One of the main barriers to the deployment of end-to-end quality of service (QoS) guarantees over the Internet is the necessity for different operators – or Autonomous Systems (AS) – to cooperate. Indeed end-users are in general not connected to the same operator and all operators involved in the communication between these two parties must promise and deliver the desired QoS.

Several mechanisms have been proposed to enable the computation of inter-domain constrained paths [3], [4], [16]. The proposed algorithms compute such paths with QoS constraints expressed with additive metrics such as delay or cost. They assume that every AS is willing to contribute to the best of their ability for the constrained path, and focus solely on the path computation issue. However, the successful establishment of a path does not mean that the different entities involved on this path will behave as announced, or that the promised QoS will be delivered. These failures can either result from unexpected changes in the underlying networks (*e.g.* a link failure or an unexpected increase in traffic) or from malicious behaviors, that is non-cooperative parties.

One possibility to detect unreliable paths is to rely on reputation mechanisms, which help identify ASes that frequently renege on the QoS they guarantee. Reputation mechanisms tend to be an effective tool to encourage trust and cooperation in distributed systems in which entities (*i.e.* users, domains, operators) exhibit a rational behavior [8]. By providing the means to rate each involved entity, a reputation mechanism aggregates these ratings, and derives publicly available reputation scores computed with a well-specified function. Hence, entities gaining a good reputation are those providing correct services to the others, and honoring the commitments they made. Reputation mechanisms have already been proposed to harness the local knowledge of a given AS in order to determine whether they can trust their neighbors [15], but, to the extent of our knowledge, none has been proposed to collect the system knowledge about reliable – and unreliable – paths.

In their highly competitive environment, it is likely that a non-negligible set of ASes will try to deceive their clients, either through collusion or not; they may also try to attract more traffic than they can reasonably provide, or gain a particular type of traffic. When an AS fails to respect its engagements, it is important that end-users be aware of such a failure in order to choose reliable routing paths with QoS promises. However, reliably and correctly distinguishing fair ASes from unfair ones is a real challenge. The reason stems from the fact that each AS has a clear incentive to put the blame on the other ASes in order to avoid the negative consequences of its failures. Even though each AS corresponds to a well-identified company, unfair behaviors do exist. Therefore, providing a usable mechanism capable of discouraging ASes from promising attractive QoS guarantees that will not be respected, and preventing ASes from shifting the blame for their own failures to others is mandatory for the welfare of inter-domain routing.

In this article, we present the main features reputation mechanisms can offer to inter-domain QoS routing in order to produce an environment where trust thrives between ASes and end-users. Our solution relies on a combination of different research areas. Specifically, Section II introduces the current state-of-the-art of building network path between several domains, while guaranteeing QoS constraints. Section III presents the required properties when using a reputation mechanism in the aforementioned context. Section IV provides the coarsed-grain design of such a reputation mechanism. Finally, we discuss threats and perspectives of reputation mechanisms in Section V.

II. Mechanism to Compute the Inter-Domain QoS-Guaranteed Path

The ever-growing appetite of users for watching videos, doing video conferences, or even gaming over the Internet stresses network capacities. Cisco pointed out in a recent study that video will represent 80 % of the traffic on Internet in 2019 [6]. This evolution stresses the need for QoS guarantees

This work was partially funded by the French ANR project AMORES (ANR-11-INSE-010).

for a non-negligible portion of Internet communications. Although mechanisms have been standardized and are already deployed, there is still no end-to-end QoS offers. One important obstacle lies in the structure of the Internet, which is composed of a multitude of independent and rational ASes that, so far, do not have sufficient incentives to cooperate in order to create end-to-end QoS offers.

The simple interaction model that has been considered so far is the following one: an application or a user formulates a request for a QoS-guaranteed path between a source S and a destination D. Besides, ASes propose QoS offers, expressed as a vector of metrics, that they are able to satisfy between a pair of their border routers. The inter-domain mechanism is distributed and combines those offers to provide possible end-to-end QoS-guaranteed paths between the source and the destination, which can then be selected by the source. This model has a limitation: in order to provide end-to-end QoS-guaranteed paths between a source S and a destination D, every AS involved in those paths must participate and announce QoS offers. To ensure that a significant number of ASes are willing to participate, a revenue is associated to the fulfillment of the request. The issues on revenue sharing between the ASes are addressed in [1] and [7].

Authors of [9] and [10] propose a mechanism allowing to find an end-to-end QoS-guaranteed path within the Internet without requiring to identify the sequence of ASes between S and D beforehand. The Internet being currently made of more than 50,000 ASes, it is not conceivable to solicit each AS to identify all the feasible paths. Therefore, the authors propose to limit the path exploration by considering all the ASes located in the neighborhood of a given path between S and D (*e.g.* the regular BGP path between S and D). They define the neighborhood and show the scalability of their mechanism, while taking advantage of the path diversity offered by the enlargement of the considered subset of ASes.

Specifically, the computation of the path is performed in a distributed manner by the Path Computation Elements (PCE) available in each AS. The main interest of a PCE is its knowledge of the routing plan of the AS, its ability to perform complex optimization calculations and to communicate with other PCEs, located either in the same AS or in neighboring ASes. The computation is made of three main phases. In the first phase, S sends the request to its PCE, that will forward it to the PCE in the destination AS. In the second phase, the PCE in D's domain initiates the possible path computation. At the end of this phase, the source receives a tree representing the concatenated candidate paths from S to D. Note that for confidentiality issue, only the AS border routers are represented in the structure. In the third phase, the source enumerates the possible path and chooses one according to its own criterion; for instance, in the network described in Figure 1, the possible paths could be $(\mathsf{AS}_1, \mathsf{AS}_4, \mathsf{AS}_6, \mathsf{AS}_8)$ and $(\mathsf{AS}_1, \mathsf{AS}_2, \mathsf{AS}_4, \mathsf{AS}_5, \mathsf{AS}_8)$. This concludes the path computation. Although the failure of the path establishment is an acceptable outcome (*e.g.* when the needed resources are no longer available), a failure after the path establishment, *i.e.* during its exploitation, is unacceptable and has to be avoided.

The following section describes the properties required to integrate a reputation mechanism between the path establishment and its exploitation.

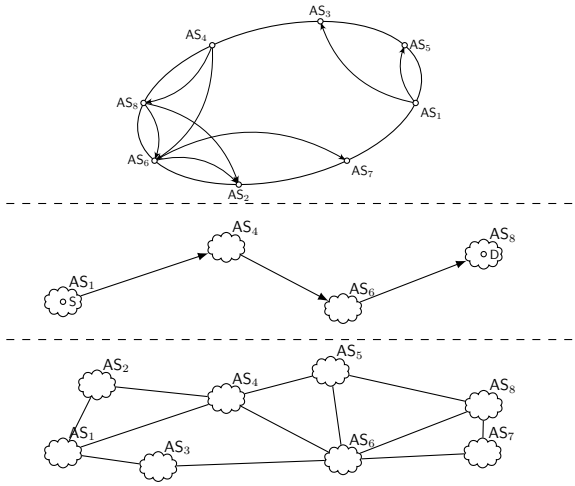

Figure 1. Network layers of our proposal. The first layer corresponds to the existing links between the ASes, the second one to the chosen path $\prec \mathsf{S}, \mathsf{D} \succ$. The final layer corresponds to the DHT organizing the score managers of ASes; AS_8 and AS_6 are the score managers of AS_4, *etc*.

III. Which Properties Should Be Provided by a Reputation Mechanism to Cope with Inter-Domain QoS Routing Features?

Reputation mechanisms have proven to be useful in multiple contexts, including electronic commerce applications [18]. These mechanisms provide a reputation score to each provider based on its past experiences. In our context, reputation scores should help end-users to select the right path of ASes to reach the desired destination with the required QoS. A reputation score is a value that reflects the opinion of clients during their past transactions with a provider. Different pieces of information can be taken into account by the reputation mechanism to compute precise reputation scores, including for example the importance given to recent feedback with respect to old ones (see [11] for a comprehensive survey of reputation score computation functions).

A. System Model

The system is composed of entities that can be of two types: end-users, *i.e.* sources and destinations, and ASes. In the following, we denote by S and D the two involved end-users. We suppose that S and D have computed a path consisting of n ASes $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$, and that they wish to compute this path' reputation before using it. Once this path has been used, they want to rate it to testify whether the ASes delivered the promised QoS or not. We call *transaction* the path exploitation between S and D for which the ASes promised QoS guarantees; we call *interaction* the whole exchange which allows S to verify the reputation and D to rate the path after the transaction.

The reputation of the path is a combination of the reputation scores of its ASes: it is computed from all the ratings issued on the ASes. In this work, we do not specify the reputation function used to compute reputation scores: our reputation mechanism may use any such function, for instance the one proposed by Anceaume *et al.* [2].

After the message exchange, the destination is able to

observe the QoS delivered, and to compare it with the promised one. Note that D is not able to determine which AS – possibly ASes – did not respect their commitment. The only information is that the whole path did not deliver the promised QoS. Therefore, if the actual QoS delivered is not sufficient, D must issue the *same* negative rating on all the ASes of the path, and the *same* positive one if the delivered QoS was correct.

B. Security Properties

An important aspect of reputation mechanisms is their aptitude to defend against various types of attacks; one of the most severe one is the manipulation of ratings and, worse, of reputation scores. Vulnerability to such attacks severely discredits the soundness of reputation mechanisms. Defending against those attacks comes down to guaranteeing the *unforgeability* of ratings and reputation scores, as well as to ensuring the *undeniability* of ratings. Briefly, the unforgeability of ratings states that an entity cannot arbitrarily create (*i.e.* "forge") a rating that is not related to a legitimate transaction, while the unforgeability of a reputation score means that such a score is computed exclusively from the set of ratings received by the concerned AS. The undeniability of ratings stipulates that, if a couple $\prec \mathsf{S}, \mathsf{D} \succ$ computes and exploits a given path, then D must be allowed to issue a rating, in the sense that no AS on the path can intentionally block D's rating. More formally, these security properties are expressed as follows.

Property 1 (Ratings unforgeability). *Consider a rating r from $\prec \mathsf{S}, \mathsf{D} \succ$ on $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$. If at least one of those $(n+2)$ entities is honest, then this rating was issued from a legitimate transaction, and was chosen by* D.

Property 2 (Reputation scores unforgeability). *The reputation score exhibited by an AS is computed exclusively from every rating this AS has received so far.*

Property 3 (Ratings undeniability). *If $\prec \mathsf{S}, \mathsf{D} \succ$ computes and exploits the path composed of $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$, then* D *is* always *able to issue a rating on $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$.*

Those three properties guarantee that the reputation mechanism uses legitimate ratings to compute the reputation scores; ASes can then use those reputation scores without modifying them. However, the reputation engine itself is a target of attacks. Dellarocas presents two major attacks against them, called *ballot-stuffing* and *bad-mouthing* [8]. To carry out a ballot-stuffing attack, a client issues numerous positive ratings about a single provider in order to increase this provider's reputation; bad-mouthing attacks are similar, except that clients issue negative ratings instead of positive ones. A solution to prevent such attacks is to guarantee the *linkability* of ratings [5], [14], which allows to detect that two ratings issued on a given provider were issued by the same client. The reputation engine can thus detect a single client issuing numerous ratings on a given provider, and can appropriately manage the ratings of the ballot-stuffing attack. This is described by Property 4. In our case, the ratings are issued by two clients: the source and the destination; instead of only the client, the couples $\prec \mathsf{S}, \mathsf{D} \succ$ must thus be publicly linkable.

Property 4 (Ratings linkability). *Two ratings from the same couple $\prec \mathsf{S}, \mathsf{D} \succ$ on a given path are publicly linkable.*

We previously explained that end-users cannot know which ASes managed to deliver the promised QoS, and which failed to do so. Consequently, we must guarantee that a couple of end-users *consistently* rates the ASes of the used path: all ASes must receive the same rating.

Property 5 (Ratings consistency). *Consider a couple $\prec \mathsf{S}, \mathsf{D} \succ$ exploiting a path $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$ and issuing ratings $(r_1, \ldots, r_n)$. Then, the value of all the ratings must be identical, that is*

$$r_i = r_j, \quad \forall 1 \leqslant i, j \leqslant n.$$

IV. A Reputation Mechanism for Inter-Domain QoS Routing

We now present the main design principles that a reputation mechanism should provide to allow a source and a destination to choose an inter-domain path based on both the promised QoS and the reputation of the path. We first present a solution guaranteeing the integrity of ratings, and briefly describe solutions to the security properties. We finally detail an interaction between a source, a destination, and the chosen ASes.

A. Securely Storing Ratings

Guaranteeing the unforgeability of ratings (Prop. 1) is an essential feature of reputation mechanisms to guarantee that no illegitimate ratings can be issued. However, this property does not ensure the integrity of ratings once they have been issued. More precisely, the storage of ratings must be secure. Similarly to EigenTrust [12] and to the mechanism proposed by Anceaume *et al.* [2], we preconize to rely on a Distributed Hash Table (DHT) to guarantee the integrity of ratings and allow entities to efficiently contact a given entity's neighbors.

More precisely, we assume that the ASes are randomly organized in a DHT like Chord [19]. The k closest neighbors of a given AS are its *score managers*: they store the ratings concerning this AS – that is, the j-th score manager of AS_i stores the set $\mathcal{R}_{i,j}$ – and send it to any querying end-user. A querying end-user accepts a set of ratings $\mathcal{R}_{i,j}$ only if at least $\lceil k/2 \rceil$ score managers have returned it. Thus, in order to modify an AS's ratings, a collusion must include at least a majority of this AS's score managers. Therefore, the DHT allows to efficiently and securely obtain the ratings concerning a given AS. Figure 1 describes the three network layers: the links between ASes, the path chosen by $\prec \mathsf{S}, \mathsf{D} \succ$, and the DHT; to obtain the reputation of path $(\mathsf{AS}_1, \mathsf{AS}_4, \mathsf{AS}_6, \mathsf{AS}_8)$, S contacts the score managers of those four ASes, that is AS_2, AS_3, AS_5, AS_6, AS_7, and AS_8.

The number of necessary score managers depends on the system parameters and on the desired maximal probability of collusion. For instance, consider a system of 50,000 ASes, including $5\ \%$ of malicious ones, where a collusion of half or more score managers may modify an AS's reputation; in this case, guaranteeing a probability of collusion less than 2^{-64}, requires for each AS to have 49 score managers. This number is given by the hypergeometric distribution, and this analysis is detailed in [13, Appendix A].

B. Guaranteeing the Security Properties

In order to guarantee the security properties previously defined, classical solutions include signature schemes for both the

unforgeability and the undeniability of ratings and reputation scores. Signature schemes allow entities to sign messages, and guarantee that no one except the legitimate owner of a (secret) signing key sk is able to output a signature on any message, which is valid with the corresponding (public) verification key vk.

More specifically, in order to guarantee the unforgeability of ratings, each entity involved in a transaction – that is, S, D, and the ASes $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$ – must all sign a random identifier of the transaction. Since signatures are unforgeable, the adversary cannot output valid illegitimate ratings. To guarantee that the destination's rating is not modified, D must additionally sign it.

The undeniability of ratings states that the destination must always be able to rate the ASes after the transaction, whatever the malicious behavior of ASes. As explained previously, issuing a rating requires signatures from the involved ASes on an identifier of the transaction. Hence, to ensure Property 3, the destination must have access to the ASes' signatures after the transaction.

Since the source and the destination are not anonymous, linking the ratings is easy: if multiple ratings were issued by a given couple $\prec \mathsf{S}, \mathsf{D} \succ$ on the same AS, the reputation engine can keep only the latest ratings from this couple in order to prevent ballot-stuffing or bad-mouthing attacks.

The consistency of ratings means that when a couple $\prec \mathsf{S}, \mathsf{D} \succ$ rates a path $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$, the ratings on all ASes must have the same value. To guarantee this consistency property, the score managers of all ASes involved in an interaction can synchronize between themselves using a consensus algorithm [17] to ensure that they all accept the same rating.

C. Setup of the Entities

During an interaction, entities need cryptographic keys and identifiers. We also assume a Public Key Infrastructure (PKI) which certifies those keys and identifiers. Specifically, end-users need an identifier id_u and a signature key pair $(\mathsf{sk}_u, \mathsf{vk}_u)$, which are both certified with cert_u, while the ASes possess an identifier and a signature key pair, also certified: AS_i has $\mathsf{id}_{\mathsf{AS}}$ and $(\mathsf{sk}_{\mathsf{AS}_i}, \mathsf{vk}_{\mathsf{AS}_i})$, certified with $\mathsf{cert}_{\mathsf{AS}_i}$. Any signature scheme can be used.

In the following, we assume that entities have authenticated each other using the PKI; that is, they know each other's public key and can thus verify the signatures.

D. Reputation Mechanism

An interaction between S, D, and $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$ proceeds in four steps. First, S contacts the score managers of each of the n ASes to obtain their ratings and computes the reputation score. The source and the destination then prepare themselves for the transaction by guaranteeing that D will be able to issue a rating. After the transaction, D is thus able to rate the ASes. Finally, the score managers of the n ASes synchronize themselves to take the rating into account.

As explained in Section IV-A, S contacts the score managers of each of the ASes of path $(\mathsf{AS}_1, \ldots, \mathsf{AS}_n)$ to compute its reputation score. The score managers of AS_i reply with their set of ratings $\mathcal{R}_{i,j}$ issued on AS_i, which they sign in order to prevent any modification. S accepts a set of ratings $\mathcal{R}_i$ as soon as half or more score managers have returned it. S then computes the reputation of the path from $\mathcal{R} = \bigcup_{1 \leqslant i \leqslant n} \mathcal{R}_i$; any reputation engine can be used to do so. Once S has computed the reputation scores of all available paths, it can choose the one to use.

The preparation to the transaction is done in three steps, as described in Figure 2.

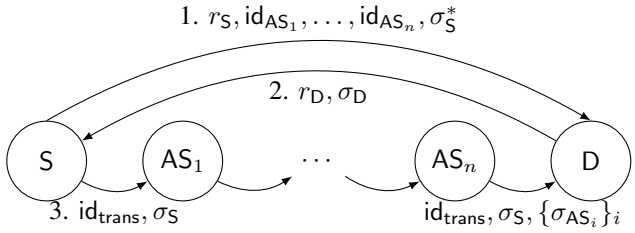

Figure 2. Preparation to the transaction

First, S chooses a nonce r_{S} – *i.e.* a random number different for each transaction –, signs this nonce and the identifiers of the $(n+2)$ entities by computing

$$\sigma^*_{\mathsf{S}} = \mathrm{Sign}\big(\mathsf{vk}_{\mathsf{S}}, H(\mathsf{id}_{\mathsf{S}}, \mathsf{id}_{\mathsf{D}}, \mathsf{id}_{\mathsf{AS}_1}, \ldots, \mathsf{id}_{\mathsf{AS}_n}, r_{\mathsf{S}})\big),$$

where H is a hash function such as SHA-256, and sends the nonce, the chosen path $(\mathsf{id}_{\mathsf{AS}_1}, \ldots, \mathsf{id}_{\mathsf{AS}_n})$, and the signature to the destination. If the signature is valid, D chooses its own nonce r_{D}, computes the identifier of transaction $\mathsf{id}_{\mathsf{trans}} = H(\mathsf{id}_{\mathsf{S}}, \mathsf{id}_{\mathsf{D}}, \mathsf{id}_{\mathsf{AS}_1}, \ldots, \mathsf{id}_{\mathsf{AS}_n}, r_{\mathsf{S}}, r_{\mathsf{D}})$, and signs it with $\sigma_{\mathsf{D}} = \mathrm{Sign}(\mathsf{vk}_{\mathsf{D}}, \mathsf{id}_{\mathsf{trans}})$. D then sends those elements to the source. If σ_{D} is valid, S computes the identifier of transaction and signs it with $\sigma_{\mathsf{S}} = \mathrm{Sign}(\mathsf{vk}_{\mathsf{S}}, \mathsf{id}_{\mathsf{trans}})$. The source then sends $\mathsf{id}_{\mathsf{trans}}$ and σ_{S} to the first AS, which also signs the identifier of transaction with $\sigma_{\mathsf{AS}_1} = \mathrm{Sign}(\mathsf{vk}_{\mathsf{AS}_1}, \mathsf{id}_{\mathsf{trans}})$. It then sends $\mathsf{id}_{\mathsf{trans}}$ and the two signatures to the second one, and similarly until the message has reached AS_n. This AS finally sends $\mathsf{id}_{\mathsf{trans}}$, σ_{S} and the ASes' signatures $\{\sigma_{\mathsf{AS}_i}\}_i$ to D. Once D has confirmed it to S, they can proceed to the transaction itself.

After the transaction, D can observe the provided QoS and issue a rating, which is positive if this QoS is consistent with the promised one, and negative otherwise. To do so, D chooses a rating ρ and signs it with $\sigma_\rho = \mathrm{Sign}\big(\mathsf{sk}_{\mathsf{D}}, H(\mathsf{id}_{\mathsf{trans}}, \rho)\big)$. Finally, D sends the following elements to the score managers of the n ASes:

$$\mathsf{id}_{\mathsf{S}}, \mathsf{id}_{\mathsf{D}}, \{\mathsf{id}_{\mathsf{AS}_i}\}_i, \mathsf{id}_{\mathsf{trans}}, \sigma_{\mathsf{S}}, \{\sigma_{\mathsf{AS}_i}\}_i, \sigma_\rho.$$

With those elements, D is able to prove that (i) the source chose the path of ASes with σ_{S}, (ii) the ASes agreed to route the message, with the signatures $\{\sigma_{\mathsf{AS}_i}\}_i$ on $\mathsf{id}_{\mathsf{trans}}$, (iii) the destination chose the rating ρ, with σ_ρ.

Finally, the ASes' score managers synchronize themselves to take this rating into account. As explained in Section IV-B, all the score managers of the n involved ASes synchronize to ensure that D issues a unique rating on the n ASes.

V. Discussions and Perspectives

In this article, we proposed a method allowing end-users to evaluate to what extent they can trust the QoS promised by

ASes, and thus to better evaluate the reliability of routing paths. The reputation mechanism proposed is directly integrated within the path computation. Furthermore, our proposition is scalable thanks to the use of efficient tools, such as the DHT, and the overhead is low: each user only needs to sign few messages.

Our proposal associates reputation scores to routing paths in order to help end-users. However, since end-users cannot observe the behavior of individual ASes, reputation scores cannot be extremely precise: if an AS is faulty, every AS of the used path receives a negative rating, which is not necessarily representative of their behavior. Improving this requires measuring elements that would be able to detect precisely which AS did not deliver the promised QoS, and which did. Thus reputation mechanisms could also assist inter-domain QoS routing to decide whether it is necessary to maintain a link with a neighboring AS or not. Integrating a double reputation mechanism – both for end-users and for ASes – would greatly improve the efficiency of inter-domain QoS routing.

REFERENCES

[1] Isabel Amigo, Pablo Belzarena, Federico Larroca, and Sandrine Vaton. Network bandwidth allocation with end-to-end QoS constraints and revenue sharing in multi-domain federations. In *Proceedings of the 7th International Workshop on Advanced Internet Charging and QoS Technology (ICQT)*, pages 50–62, 2011.

[2] Emmanuelle Anceaume, Gilles Guette, Paul Lajoie-Mazenc, Nicolas Prigent, and Valérie Viet Triem Tong. A privacy preserving distributed reputation mechanism. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1951–1956, 2013.

[3] Gilles Bertrand, Samer Lahoud, Miklós Molnár, and Géraldine Texier. *Inter-Domain Path Computation with Multiple QoS Constraints*, chapter 8, pages 129–147. Nova Science Publisher, 2010.

[4] Gilles Bertrand, Samer Lahoud, Géraldine Texier, and Miklós Molnár. A distributed exact solution to compute inter-domain multi-constrained paths. In *Proceedings of the 15th Open European Summer School and IFIP TC6.6 Workshop (EUNICE)*, 2009.

[5] John Bethencourt, Elaine Shi, and Dawn Song. Signatures of reputation. In *Proceedings of Financial Cryptography and Data Security (FC)*, pages 400–407, 2010.

[6] Cisco Visual Networking Index. White paper: Forecast and methodology, 2014–2019, 2015.

[7] Costas Courcoubetis, Manos Dramitinos, George D. Stamoulis, G. Blocq, Avi Miron, and Ariel Orda. Inter-carrier interconnection services: QoS, economics and business issues. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2011.

[8] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 150–157, 2000.

[9] Romain Jacquet, Geraldine Texier, and Alberto Blanc. Sanp: An algorithm for selecting end-to-end paths with QoS guarantees. In *Proceedings of the IEEE Future Network and Mobile Summit Conference (FutureNetworkSummit)*, pages 1–10, 2013.

[10] Romain Jacquet, Geraldine Texier, and Alberto Blanc. Computing end-to-end QoS paths in the internet considering multiple alliances. In *Proceedings of the International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, pages 1–6, 2014.

[11] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[12] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 640–651, 2003.

[13] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent, and Valérie Viet Triem Tong. Efficient distributed privacy-preserving reputation mechanism handling non-monotonic ratings. Technical Report https://hal.archives-ouvertes.fr/hal-01104837, IRISA, 2015.

[14] Paul Lajoie-Mazenc, Emmanuelle Anceaume, Gilles Guette, Thomas Sirvent, and Valérie Viet Triem Tong. Privacy-preserving reputation mechanism: A usable solution handling negative ratings. In *Proceedings of the IFIP WG 11.1 International Conference on Trust Management*, pages 92–108, 2015.

[15] Mohamed Lamine Lamali, Dominique Barth, and Johanne Cohen. Reputation-aware learning for sla negotiation. In *Proceedings of the NETWORKING 2012 Workshops*, pages 80–88. 2012.

[16] N. Le Sauze, A. Chiosi, R. Douville, H. Pouyllau, H. Lonsethagen, P. Fantini, C. Palasciano, A. Cimmino, Peter Reichl, and Ivan Gojmerac. ETICS: QoS-enabled interconnection for future internet services. In *Proceedings of the IEEE Future Network and Mobile SummitCconference (FutureNetworkSummit)*, 2010.

[17] Michel Raynal. Consensus in synchronous systems: A concise guided tour. In *Proceedings of the Pacific Rim International Symposium on Dependable Computing*, pages 221–228, 2002.

[18] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[19] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, Marinus Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, 2003.